

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants: C. Lupu et al. Attorney Docket No.: MSFT114614
Application No.: 09/544,512 Group Art Unit: 2124
Filed: April 6, 2000 Examiner: Tuan A. Vu
Title: FIXING INCOMPATIBLE APPLICATIONS USING A LIGHT DEBUGGER

AMENDMENT TRANSMITTAL LETTER

Seattle, Washington 98101

RECEIVED

April 7, 2003

APR 14 2003

TO THE COMMISSIONER FOR PATENTS:

Technology Center 2100

A. Amendment Transmittal

Transmitted herewith is an amendment in the above-identified application.

- X 1. No additional claim fee is required, as shown below.
_____ 2. The claim fee has been calculated as shown below.

COMPUTATION OF FEE FOR CLAIMS AS AMENDED

	Claims Remaining After Amendment		Highest Number Previously Paid For		Present Extra		Rate		Additional Fee
Total Claims	18	-	20	=	0	x	18	=	0
Independent Claims	3	-	3	=	0	x	84	=	0
TOTAL									\$0

B. Additional Fee Charges or Credit for Overpayment

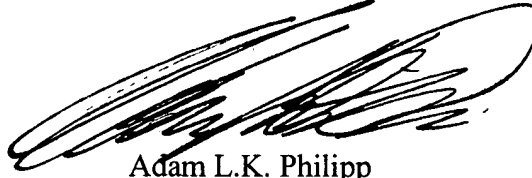
The Commissioner is hereby authorized to charge any fees under 37 C.F.R. §§ 1.16, 1.17 and 1.18 which may be required during the entire pendency of the application, or credit any overpayment, to Deposit Account No. 03-1740. This authorization also hereby includes a request

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PLLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

for any extensions of time of the appropriate length required upon the filing of any reply during the entire prosecution of this application.

Respectfully submitted,

CHRISTENSEN O'CONNOR
JOHNSON KINDNESS^{PLLC}

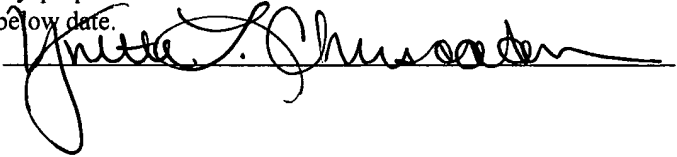


Adam L.K. Philipp
Registration No. 42,071
Direct Dial No. 206.695.1792

I hereby certify that this correspondence is being deposited with the U.S. Postal Service in a sealed envelope as first class mail with postage thereon fully prepaid and addressed to the U.S. Patent and Trademark Office, P.O. Box 2327, Arlington, VA 22202, on the below date.

Date:

April 7, 2023



ALP:ylc

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PLLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100



BOX NON-FEE
AMENDMENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants: C. Lupu et al.

Attorney Docket No.: MSFT114614

Application No.: 09/544,512

Group Art Unit: 2124

Filed: April 6, 2000

Examiner: Tuan A. Vu

Title: FIXING INCOMPATIBLE APPLICATIONS USING A LIGHT DEBUGGER

RESPONSE AND REQUEST FOR RECONSIDERATION

Seattle, Washington 98101

April 7, 2003

RECEIVED

APR 14 2003

Technology Center 2100

TO THE COMMISSIONER FOR PATENTS:

This communication is in response to the Office Action ("Office Action") mailed on January 9, 2003, in which Claims 1-18 were rejected. Claims 1-18 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Preisler et al. (U.S. Patent No. 5,675,803) ("Preisler") in view of Kalwitz et al. (U.S. Patent No. 5,815,722) ("Kalwitz"). Applicants respectfully submit that this rejection is in error, should be withdrawn and the application allowed.

Prior to discussing the reasons why applicants believe that the claims in this application are allowable, a brief discussion of the present invention, followed by a brief discussion of the cited and applied references, is presented. The following discussion of applicants' invention and the cited and applied references are not provided to define the scope or interpretation of any of the claims of this application. Instead, these discussions are provided to help the United States Patent and Trademark Office better appreciate important claim distinctions discussed thereafter.

Summary of the Invention

The present invention is directed to providing a system, method and computer-readable medium for patching applications that would not otherwise be compatible with the operating system on which the application is run. In one embodiment of the present invention, a debugger

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PLLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

is run that executes and appropriately patches the incompatible application. The debugger application loads a dynamic link library (DLL) containing patches for the functions that are incompatible with the operating system. If an application runs properly under the operating system, the debugger application and associated DLL is not loaded. Therefore, applications that work correctly under the operating system are not required to go through an additional level of execution of the debugger application.

One exemplary embodiment of the invention works in the following way. First, the user starts an application. Second, the application is identified as being compatible or incompatible with the operating system. To determine whether or not the application is compatible with the operating system, identifying information from the application is checked against a database containing a list of applications that are currently known to be incompatible. If the name of the application is found to be contained within the database, a set of attributes are checked to see if the particular version of the application is incompatible. If all the attributes that are checked match the ones in the database, the application is found to be incompatible with the current operating system. If the application name is not found in the database, the application is found to be compatible with the current operating system. Third, the application is either run with a debugger or directly without a debugger, depending on the application's compatibility with the operating system.

If the application is found to be incompatible with the operating system, the operating system starts a debugger application that, in turn, runs the incompatible application. Before loading the incompatible application, the debugger loads a DLL that contains patches for the incompatible functions within the application. Specifically, the DLL contains a list of breakpoints specifying the location where the application needs to be patched together with the appropriate code required to patch the application.

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{LLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

When the debugger loads the DLL, a function is called that sets the breakpoints within the incompatible application. A breakpoint may be specified by: (1) the module name and function name; (2) the module name and a memory offset; or (3) an address. The debugger also implements a handler that is capable of modifying the application code that is incompatible. In one embodiment of the invention, the debugger handler is capable of reading and writing memory within the incompatible application. Therefore, the handler is capable of modifying the incompatible code or inserting code into the application to fix the problem code contained within the application. For example, when the debugger reaches a breakpoint within the application, a handler may be called that merely skips a portion of the incompatible code or the handler may rewrite a portion of code to contain a certain value in order to make the application compatible with the operating system.

One of the benefits of the use of a debugger application to patch incompatible applications is that it is very robust. The debugger is capable of monitoring every step an application takes while it is executing. Additionally, the amount of code required to patch an application is generally very small, such as 200 bytes. This small amount of code makes patches easily accessible to a user either through a website or FTP site.

Summary of the Principal References Cited

The Preisler Reference

Preisler describes a part of a run-time debugger operation designated a "Fix and Continue" operation. Preisler permits a user to begin a debugging session if an error is encountered in the code. The user edits the corresponding source code to correct the error and then executes a "Fix and Continue" command without leaving the debugging session. The Fix and Continue command calls a compiler to recompile the source code file with the newly edited text, receives the resulting recompiled object code file from the compiler, uses a dynamic linker

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PLLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

to link the recompiled object code into the target application program process, patches the previous version of this same object code file to refer to the newly recompiled code, resets any required variables and registers, resets the program counter to the line of code being executed when the error was discovered. The debugger then continues the debug session thereby saving the time it would ordinarily take to quit the debug session, relink and reload the target program and start the debug session once again.

Preisler's teaching of a manual "Fix and Continue" command (see col. 2, ll. 51-55) merely provides for pausing the debugging of a program to allow a programmer to edit source code without leaving the debugger. Preisler nowhere teaches or even remotely suggests adding patches into a program based on a determination that an application is incompatible with an operating system. As correctly pointed out in the Office Action, Preisler has no mechanism for detecting an operating system incompatibility. Nor does Preisler teach or even remotely suggest breakpoint handlers with instructions for patching programs at debugger breakpoints.

The Kalwitz Reference

Kalwitz teaches a system and apparatus for altering an executable file stored in a random access memory ("RAM") on a designated interactive network board by activating a local area network ("LAN") communication program. The communication program operates to broadcast an inquiry through the LAN for the particular network board, to receive location information of the network board in response to the broadcast inquiry, and to establish communication with the network board. The executable file is downloaded into RAM on the network board through a LAN interface. Execution of the executable file may then be commanded remotely, e.g., across the LAN interface.

Kalwitz has no teaching or suggestion of testing for operating system compatibility, nor any analogous software instructions. Rather, Kalwitz teaches a hardware network expansion

board ("NEB") that may load software to configure its printing and network compatibilities. However, even with the NEB, Kalwitz has no operating system compatibility detection functionality.

The only determining for compatibility in Kalwitz is prescanning for communication protocol matching. One of ordinary skill in the art will appreciate that a communication's protocol is not (and is not analogous to) an operating system. Merely broadcasting a scan on a LAN to determine what communication protocols are in use, is not analogous to, and does not teach or suggest determining operating system compatibilities.

The Claims Distinguished

The Office Action has failed to show and the applicants are unable to find, where any of the cited and applied references, either alone or in combination, disclose the subject matter of the claimed invention. Independent Claims 1, 7, and 13, the only independent claims in this application read as follows:

1. A method for patching a computer application program including a plurality of executable steps, comprising:

(a) *determining whether or not the computer application program is compatible with a computer operating system executing the computer application program;* and

(b) *if the computer application program is determined to be incompatible with the computer operating system,* starting a debugger to run the computer application program.
(Emphasis added)

7. A computer-readable medium having computer executable instructions for patching a computer application program including a plurality of executable steps, which, when the computer executable instructions are executed, comprise:

(a) *determining whether or not the computer application program is compatible with a computer operating system executing the computer application program;* and

(b) *if the computer application program is determined to be incompatible with the computer operating system,* starting a debugger to run the computer application program.
(Emphasis added)

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{LLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

13. A computer system for patching a computer application program, wherein the computer system is capable of running an application having a plurality of executable steps, the method comprising:

- (a) *determining whether or not the computer application program is compatible with a computer operating system executing the computer application program; and*
- (b) *if the computer application program is determined to be incompatible with the computer operating system,* starting a debugger to run the computer application program.
(Emphasis added)

Among other differences, none of the applied and cited references teach "determining whether or not the computer application program is compatible with a computer operating system executing the current application program," as recited in independent Claims 1, 7, and 13. While Kalwitz' protocol scanning purportedly detects network protocols, this is not a teaching or even a remotel suggestion of the solution recited in Claims 1, 7, and 13. Thus, applicants submit that Claims 1, 7, and 13 and all of the claims dependent therefrom (2-6, 8-12, and 15-18) are allowable.

Applicants further submit that the dependent claims are allowable for reasons in addition to the reasons why the independent claims are allowable. For example, applicants have been unable to find where any of the cited and applied references discuss or even remotely suggest a debugger capable of running the incompatible application and "patching the application when a breakpoint has been reached," as recited independent Claims 2, 8, and 14. As explained above, the Preisler reference requires a programmer to manually edit source code software for its Fix and Continue system to work. Because neither Preisler nor Kalwitz teach or suggest patching the application when a breakpoint has been reached, these references taken either alone or in combination would not have rendered the subject matter of Claims 2, 8, and 14 obvious to a person of ordinary skill in the art at the time this invention was made.

Similarly, neither Preisler nor Kalwitz teach, disclose or suggest determining if an application is compatible or incompatible by: (a) "determining if at least one identifying attribute of a plurality of identity attributes of a stated application matches at least one identifying attribute of a plurality of identifying attributes of incompatible applications; and (b) if at least one of the identifying attributes matches, determining that the running application is incompatible, otherwise, determining that the application is compatible," as recited in dependent Claims 3, 9, and 15. Kalwitz teaches nothing equivalent. While Kalwitz does disclose error checking using a checksum value for Read Only Memory firmware (See Fig. 4), Kalwitz has no teaching or suggestion of matching attributes of an application to attributes of known incompatible applications. Additionally, the bit pattern matching of Kalwitz (See Fig. 8) has no relation to determining application compatibility with an operating system. The bit matching merely provides for bit-wise selection of EPROM-resident software modules using a binary configuration code (col. 27, ll. 18-21). Also, as noted above, the prescanning of a network to determine network communication protocols has no teaching or suggestion of determining application incompatibility with an operating system. Thus, applicants submit that Claims 3, 9, and 15 are allowable for reasons in addition to the reasons why the claims from which these claims depend are allowable.

The Office Action appears to assert that the Kalwitz teaching of storing a list of applications compatible with execution on a piece of hardware also teaches or suggests the subject matter recited in Claims 4, 10, and 16. Applicants respectfully disagree, particularly when the subject matter of Claims 4, 10 and 16 is considered in combination with the subject matter of the claims from which these claims depend. Kalwitz merely loads a list of compatible applications in memory. Contrariwise, Claims 4, 10, and 16 recite storing identifying attributes of incompatible applications, which is entirely different. Compatible applications have no need

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{LLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

for debugging and patching. Only incompatible applications that require debugging and patching. There is no teaching or suggestion in Kalwitz of storing the attributes of incompatible applications. Thus, applicants submit that Claims 4, 10, and 16 are allowable for reasons in addition to the reasons why the claims from which these claims depend are allowable.

The Office Action also appears to assert that the "patch area" and "patch instruction" of Preisler form a teaching or suggestion of "handlers having a set of instructions for patching the application" as recited in Claims 5, 15, and 17. Applicants respectfully disagree. It is clear that Preisler's patch area is merely a region for error checking a patch (col. 18, ll. 61-63) and the patch instruction is a call to branch to a patched area (col. 18, ll. 13-39). Preisler includes no teaching or suggestion of handlers with instructions for patching as recited in Claims 5, 11, and 17. Thus, applicants submit that Claims 5, 11, and 17 and Claims 6, 12, and 18, which respectively depend therefrom are allowable for reasons in addition to the reasons why the claims from which these claims depend are allowable.

Because the Office Action has failed to state a *prima facie* case of obviousness, applicants submit that the rejection based on the teachings of Preisler and Kalwitz is in error and should be withdrawn. Independent Claims 1, 7, and 13 are clearly patentably distinguishable over these cited and applied references. Claims 2-6, 8-12, and 14-18 are allowable because they depend from an allowable independent claim and because of the additional limitations added by these claims. Consequently, reconsideration and allowance of rejected Claims 1-18 is respectfully requested.

CONCLUSION

In view of the foregoing remarks, applicants submit that the present application is now in condition for allowance. Reconsideration and reexamination of this application, as amended, allowance of the rejected claims and passage of the application to issue at an early date is

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{LLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

respectfully solicited. If the Examiner has any questions or comments concerning this application, the Examiner is invited to contact the applicants' undersigned attorney at the number below.

Respectfully submitted,

CHRISTENSEN O'CONNOR
JOHNSON KINDNESS^{PLLC}

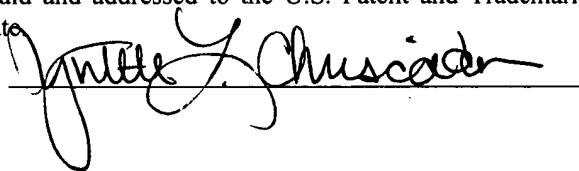


Adam L.K. Philipp
Registration No. 42,071
Direct Dial No. 206.695.1792

I hereby certify that this correspondence is being deposited with the U.S. Postal Service in a sealed envelope as first class mail with postage thereon fully prepaid and addressed to the U.S. Patent and Trademark Office, P.O. Box 2327, Arlington, VA 22202, on the below date.

Date:

April 7, 2003



ALP:ylc/mk

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PLLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100